УДК 511.9

РЕАЛИЗАЦИЯ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА ПОИСКА КРАТЧАЙШЕГО ВЕКТОРА В БЛОЧНОМ МЕТОДЕ КОРКИНА-ЗОЛОТАРЕВА

В. С. Усатюк

Братский государственный университет, г. Братск, Россия

E-mail: L@Lcrypto.com

arXiv:1304.1900v1 [cs.DM] 6 Apr 2013

В работе предложена параллельная реализация алгоритма Каннана для решения задач поиска кратчайшего и короткого векторов в решетке. Алгоритм может применяться в составе блочного метода Коркина-Золотарева, так и независимо. Эксперимент показал 3-кратное ускорение работы блочного метода Коркина-Золотарева на четырехядерной системе. С использованием алгоритма были получены 1-е и 6-е места на конкурсах поиска коротких векторов.

Ключевые слова: *решетки, паралельный алгоритм, поиск короткого вектора, поиск кратчайшего вектора, блочный метод Коркина-Золотарева.*

Определение 1. Базис $B = \{b_1, b_2, \dots, b_m\}$ решетки $L \subset R^n$ приведен блочным методом Коркина-Золотарева(BKZ, Block Korkin-Zolotarev method, [1]) блоком β, если:

1) базис B приведен по длине;
2) $\left\|b_i^{\perp}\right\| = \lambda_1(L_i), i = 1, \dots, m$, где $\lambda_1(L_i)$-кратчайший вектор в обратной (сопряженной) решетке L_i, образованной ортогональным дополнение пространства векторов $b_i, \dots, b_{\min(i+\beta-1,m)}$.

BKZ-метод содержит два основных алгоритма подлежащих распараллеливанию: ортогонализацию базиса решетки и поиск кратчайшего вектора. Вопрос распараллеливания ортогонализации базиса рассматривался в работе [2].

Решение задачи поиска кратчайшего вектора будет заключаться, в полном переборе всех линейных комбинаций векторов базиса решетки $\|x\|^2 = \|\sum_{i=1}^{m} x_i b_i\|^2 \leqslant A^2, x_i \in Z$, где A - норма искомого кратчайшего вектора. В качестве нормы берётся верхняя оценка длины кратчайшего вектора, $A = \sqrt{\gamma_m}\det(L)^{\frac{1}{m}}$, где γ_m-константа Эрмита, в тех случаях, когда наименьший из векторов в базисе решетки превосходит оценку, $\|b_1\| > \sqrt{\gamma_m}\det(L)^{\frac{1}{m}}$, [3]. По этой причине предварительное приведение базиса решетки позволяет уменьшить пространство перебора x_i. С целью уменьшения пространства перебора распишем базис решетки, через ортогональные вектора, для простоты изложения (сопряженной при вычислении на практике с утратой преимуществ параллелизма, характерных для современных QR-методов разложения) используя метод ортогонализации Грамма-Шмидта. Получим $b_i = \sum_{j=1}^{i} \mu_{i,j} b_j^{\perp}, 2 \leqslant i \leqslant m, 1 \leqslant j < i \leqslant m$, где $\mu_{i,j}$ - коэффициенты Грама-Шмидта. Легко убедиться, что в этом случае поиск кратчайшего вектора сводиться к решению системы неравенств:

$$\begin{cases} x_m^2 \left\|b_m^{\perp}\right\|^2 \leqslant A^2, \\ (x_{m-1} + \mu_{m,m-1} x_m) \left\|b_{m-1}^{\perp}\right\|^2 \leqslant A^2 - x_m^2 \left\|b_m^{\perp}\right\|^2 \\ \dots \\ (x_1 + \sum_{i=2}^{m} x_i \mu_{i,j})^2 \left\|b_1^{\perp}\right\|^2 \leqslant A^2 - \sum_{j=2}^{m} l_j \end{cases}, \text{ где } l_j = (x_j + \sum_{i=j+1}^{m} x_i \mu_{i,j})^2 \left\|b_j^{\perp}\right\|^2$$

и выбору одного из целочисленных векторов, у которого норма скалярного произведения с базисом решетки минимальна.

Формализуя задачу, получим обход дерева от корня к листу, в каждой из вершин которого решается соответствующее линейное уравнение. Из корня этого дерева выходит $2 \cdot \left\lceil \frac{A}{\left\|b_m^{\perp}\right\|} \right\rceil = 2 \cdot \left\lceil \frac{\sqrt{\gamma_m}\det(L)^{\frac{1}{m}}}{\left\|b_m^{\perp}\right\|} \right\rceil$ ветвей или $2 \cdot \left\lceil \frac{\|b_1\|}{\left\|b_m^{\perp}\right\|} \right\rceil$, в случае предварительного приведения базиса решетки. В силу симметричности дерева (по свойствам нормы), для получения искомого кратчайшего вектора нам необходимо перебрать только половину его вершин. В результате полного обхода дерева от корня к листу, мы будем получать предполагаемый кратчайший вектор x с нормой меньше либо равной искомой. Если норма полученного вектора будет меньше заданной ранее, целесообразно обновить ее, с целью уменьшения пространства перебора. Остановка алгоритма осуществляется, когда завершен обход вершин дерева или мы получили вектор с достаточной для нас нормой, в случае поиска короткого вектора. Каждый из потоков осуществляет вычисление своей ветки исходящей из корня дерева.

Алгоритм 1 Поиска кратчайшего вектора в решетке $L(B), B = \{b_1, b_2, \dots b_m\}$.

Вход: $\left\|b_1^{\perp}\right\|, \left\|b_2^{\perp}\right\|, \dots, \left\|b_m^{\perp}\right\|$, $\mu_{i,j}$, id- номер потока, начиная с 1
Выход: Вектор $x = (x_1, x_2, ..., x_m) \in Z^m : \left\|\sum_{i=1}^{m} x_i b_i\right\| = \lambda_1(L(B))$
1: $x = \{1, 0, 0, ..., 0\}, l_i = \{0\}^m, X = \{\varnothing\}$;
2: **Для** $i = 1, ..., m : l_i = (x_i + \sum_{j=i+1}^{m} x_j \mu_{j,i})^2 \left\|b_i^{\perp}\right\|^2$,
3: **Если** $(\sum_{j=i+1}^{m} l_j > A) : i = i + 1, x_i = x_i + 1$;
4: **Если** $(\sum_{j=1}^{m} l_j \leqslant A$ и $i = 1)$: **Если** $\left\|\sum_{j=1}^{m} x_j b_j\right\| < A : A = \left\|\sum_{j=1}^{m} x_j b_j\right\|$,
5: $X = X + \{x\}^m = X + \{\sum_{j=1}^{m} x_j b_j\}, x_1 = x_1 + id$;
6: **Если** $(\sum_{j=i}^{m} l_j \leqslant A$ и $i \neq 1) : i = i - 1, x_i = \left\lceil -\sum_{j=i+1}^{m}(x_j \mu_{ji}) - \frac{\sqrt{A - \sum_{j=i+1}^{m} l_j}}{\left\|b_i^{\perp}\right\|} \right\rceil$;
7: **Вывести** вектор с минимальной нормой из множества X.

Данный алгоритм был реализован на основе потоковой модели NPTL (Native POSIX Thread Library, [4]) под CentOS 6.3. Осуществляя приведение 103-мерной решетки BKZ-методом при $\beta = 52$, 4-х потоках исполняемых на AMD Phenom 965/8 Gb DDR2-800, продемонстрировал 3-кратное ускорение выполнения метода по сравнению с fplll-4.0.1, [5]. С использованием данного алгоритма были получены 1-е и 6-е места на международном конкурсе алгоритмов поиска коротких векторов [6], для норм $A = m \cdot \det(L)^{\frac{1}{m}}$ и $A = 1.05 \cdot \frac{\Gamma\left(\frac{m}{2}+1\right)^{\frac{1}{m}}}{\sqrt{\pi}} \cdot \det(L)^{\frac{1}{m}}$, соответственно.

ЛИТЕРАТУРА

1. *Schnorr C. P.* Block reduced lattice bases and successive minima // Combinatorics, Probability and Computing. 1994. V. 3. pp. 507–522.
2. *Усатюк В. С.* Реализация параллельных алгоритмов ортогонализации в задаче поиска кратчайшего базиса целочисленной решетки // Прикладная дискретная математика. Приложение. 2012. № 5. с. 120–122.
3. *Hanrot G., Stehle D.* Improved Analysis of Kannan's Shortest Lattice Vector Algorithm. // LNCS. 2007. V. 4622. pp. 170–186.
4. *Kerrisk M.* The Linux Programming Interface: A Linux and UNIX System Programming Handbook. No Starch Press, 2010. 1552 p.
5. `http://perso.ens-lyon.fr/damien.stehle/fplll/` — Приложение fplll. 2013.
6. `http://www.latticechallenge.org/ideallattice-challenge/index.php` — Ideal lattice challenge (SVP, Approx-SVP). 2012.

Usatyuk V. S. **THE IMPLEMENTATION OF THE PARALLEL SHORTEST VECTOR ENUMERATE IN THE BLOCK KORKIN-ZOLOTAREV METHOD .** This article present a parallel CPU implementation of Kannan algorithm for solving shortest vector problem in Block Korkin-Zolotarev lattice reduction method. Implementation based on Native POSIX Thread Library and show linear decrease of runtime from number of threads.

Keywords: *shortest vector problem, SVP, block Korkin-Zolotarev, BKZ, lattices, parallel algorithms.*

Усатюк Василий Станиславович — программист кафедры дискретной математики и защиты информации Братского государственного университета, г. Братск. E-mail: **L@Lcrypto.com**